

# A Markov Decision Process Model for Traffic Prioritisation Provisioning

**Abdullah Gani**  
*The University of  
Sheffield, UK*

**Omar Zakaria**  
*The Royal Holloway,  
University of London,  
UK*

**Nor Badrul Jumaat**  
*The University of  
Malaya, Kuala  
Lumpur, Malaysia*

[abg@dcs.shef.ac.uk](mailto:abg@dcs.shef.ac.uk)

[O.B.Zakaria@rhul.ac.uk](mailto:O.B.Zakaria@rhul.ac.uk)

[badrul@um.edu.my](mailto:badrul@um.edu.my)

## Abstract

This paper presents an application of Markov Decision Process (MDP) into the provision of traffic prioritisation in the best-effort networks. MDP was used because it is a standard, general formalism for modelling stochastic, sequential decision problems. The implementation of traffic prioritisation involves a series of decision making processes by which packets are marked and classified before being despatched to destinations. The application of MDP was driven by the objective of ensuring the higher priority packets are not delayed by the lower ones. The MDP is believed to be applicable in improving the traffic prioritisation arbitration.

**Keywords:** Markov Decision Process, Traffic Prioritisation, Quality of Service, Network Application Management, and Network Resource Control.

## Introduction

Network has proliferated at the tremendous speed over the last 30 years. This proliferation came together with the proliferation in the network applications which has caused to the increasingly amount of traffic in the network. Traffic was generated as a result of the application executions. These traffics that are comprised of many different types deploy network resources and without control mechanism the resource would be under strain. This effectively can degrade the performance across the network which can affect the execution of applications. As the delay build-up is continuously escalating it reduces the network application responsiveness and gradually leads to the Denial of Service (DoS). Such scenario has inspired the network designers to find an effective way of how to differentiate traffic so that different treatment can be provided.

Apart from the network application proliferation, the density of traffic in the network also plays a part in causing the network resource under strain. Many factors had been identified that positively contribute to the density of traffic; however the network size was believed to be the most significant factor. Logically, the larger of network size the more traffic it contains. This is simply be-

cause the presence of nodes is obviously associated with the traffic generations.

The need for network traffic to be given different treatments has emerged as an important feature in the networking and has become a challenge to the network designers to find the best mechanisms that can differen-

---

Material published as part of this journal, either on-line or in print, is copyrighted by Informing Science. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission from the publisher at [Publisher@InformingScience.org](mailto:Publisher@InformingScience.org)

tiate traffic flow according to the service characteristics. Originally such need has been triggered by the requirement of applications that demanding sufficient allocation of resources so that the application execution could consistently be running without interruption. This is the case especially for applications that are sensitive to delay. The need for differentiating traffic is also triggered by the requirements for implementing the firewall and providing quality of service.

All the network applications can be categorically divided into a least two lists – casual and critical list. The casual list comprises of network applications that require no special treatment in order to run. However, the critical list is the network applications that only available with the presence of special treatment. The connotation of special treatment is loosely defined and can include bandwidth reservation and traffic prioritisation. A new list of critical application has been created appending from the current list that normally comprises of time sensitive applications such as multimedia application. This list is now getting longer and longer with applications that are critical to the organisations such as accounting or databases applications are included too. These applications require service reliability so that the traffic which carries critical organisational information over the network has to protected and guaranteed.

At the transport layer all these traffic are homogenous. Packets that comprises of user data with header are equal in term of its architecture. This has caused the difficulties in distinguishing the critical and non-critical applications that enable them to be given different treatments. We believe by differentiating the traffic flow many problems that related to the quality of service can be solved especially for pre-emptive measures of controlling. For example, the provision of traffic prioritisation is only feasible when traffic can be classed into different class-types.

In this paper, we present an application of Markov Decision Process into a problem of traffic prioritisation with the main goal of improving the arbitration process that leads to a better performance. One of the attributes of better performance over the data networks is a minimal delay for a packet to reach its destination (Bertsekas & Gallager, 1991).

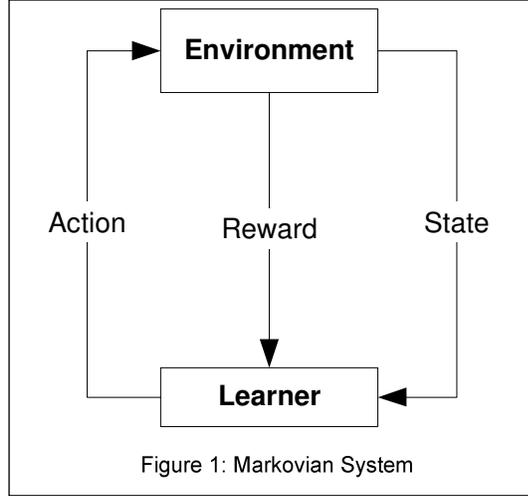
We assume that a good decision must consider the future impacts as a result of the current decision. In order to reach such good decision; it requires information of the future. The absence of perfect information can escalate the entropy accumulation and this can make such decision be capable of causing the deviation from its original goal. The traffic prioritisation is a process that involves a series of decision making and each decision can effectively affects the performance of other activities. For example, the decision of classifying the packets can affects the popping process directly because both of them are interdependence.

In the next section, we briefly present the fundamental idea of Markov Decision Process and followed by the description of traffic prioritisation problem.

## Markov Decision Process

Markov Decision Process (MDP) is a mathematic model that is useful in the study of complex systems (Howard, 1966). The connotation of complex system refers to its states that are described by the values of variables and stochastic. It is used primarily for the decision making processes. In a decision making process, choosing the best alternative (action) is not an easy task. It requires thinking about more than just the immediate effects of the action. Sometimes action with poor immediate effects can have better long term results. This can be achieved through the ‘exploration’ and ‘exploitation’ of all possible policies. In order to yield the best possible solution, MDP is an appropriate technique for modelling such problems and allowing us to automate the processes (Ratitch & Precup, 2002; Tijms Henk 1994; White 1978)

Fundamentally, MDP is characterised by 4 tuples (state, action, transition probability, reward function) and a system is said to be *Markov Property* when the effects of an action taken in a state



depend only on that state and not on the previous states. In other words, the next state of the system depends only on the present state, not on the preceding states (Sutton & Barto, 1998).

In a *Markovian system* a learner interacts with the environment through a set of actions  $a \in A$ . At every discrete time step  $t$ , the learner perceives state,  $s_t$  and performs *action*  $a_t$ . One time step later, the learner receives an immediate *reward*  $r_{t+1}$  and the environment transition to a new state,  $s_{t+1}$ . The Markov property means that the next state,  $s_{t+1}$  and the immediate reward,  $r_{t+1}$  depend only on the current state and action,  $\{s_t, a_t\}$ . The

MDP model also consists of the transition probabilities,  $p_{s,s}^a$ , and the expected values of the immediate rewards  $R_{s,s}^a$ ,  $\forall s, a, s'$ . The learner learns the policy,  $\pi$  to maximise the cumulative reward over time. A policy is a mapping  $\pi : S \times A \rightarrow [0,1]$ , where  $\pi(s, a)$  denotes the probability that the learner takes action  $a$  when the environment is in state  $s$ . Rewards in the long-term is a summation of reward sequence with discounted factor  $\sum_{t=0}^{\infty} \gamma^t r_{t+1}$ , where  $\gamma \in [0,1]$ .

In a MDP model, the goal of learner is to find an optimal policy  $\pi$  so as to maximise the expected sum of discounted rewards. The optimal policy  $\pi$  simply specifies the best action to take for each of the states. Policy is actually representation of a *state-value function*. We define

$V^\pi(s)$  to be a value of policy  $\pi$  when starting at state  $s$ . We write

$$V^\pi(s) = E_\pi[r_{t+1} + r_{t+2} + r_{t+3} + \dots | s_t = s] \tag{1}$$

The equation (1) is more realistic if discount factors are included.

$$V^\pi(s) = E_\pi[r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots | s_t = s] \tag{2}$$

where  $\gamma \in [0,1]$ . After the value function has been defined, policy  $\pi$  has to be evaluated using the equation (3) so as to find the optimal policy  $\pi$ .

$$V^\pi(s) = r(s, a) + \gamma \sum_s P(s' | s, a) V^\pi(s') \tag{3}$$

Equation (3) is also known as the Bellman equation and can be written in terms of the next state and reward functions.

$$Q^*(s) = r(s, a) + \gamma \sum_s P(s' | s, a) V^\pi(s') \tag{4}$$

Similarly we can define  $Q^\pi(s, a)$ , the *action-value function* for policy  $\pi$ , to be the expected return from starting in state  $s$ , taking action  $a$ , and thereafter following policy  $\pi$ . We express  $Q^\pi$  as

$$Q^\pi(s, a) = E_\pi[r_t | s_t = s, a_t = a] \tag{5}$$

We can use  $Q^*$  and  $V^*$  respectively to symbolise the state-value and action-value functions for the optimal policy  $\pi^*$ .

## Traffic Prioritisation Problem

In the recent years, the need for providing differential treatment to the network traffic has emerged as a prevalent necessity and has grasped the attention of many those in the network community (Sharda, 1999). The dark side of an increasingly volume of network traffic is that it brought in some negative impacts to the network. Packets have to compete rigorously for resource in order to reach destinations without severe delay and the failure to do so will result in the packets are likely to be discarded. An efficient way of dealing with these situations has to be found so as to improve the ‘delivery’ time of packets to destination. Traffic prioritisation was believed to be a sound solution that can ensure high priority traffic is forwarded without being delayed by lower priority traffic in a network (Hardy & NetLibrary Inc. 2001).

The definition of high priority traffic is based on the characteristics of applications which require different treatment from the normal traffic. An example of critical application is a financial application that is used in the Account department. This type of application has the needs of immediate access to large files and spreadsheets remotely. Non-critical network applications such as email which normally require no differential treatment can be regarded as critical application if the user is important to the organisation. This type of criticality is away from the standard definition despite still using the same protocol. In short, the criticality of application can be defined by a number of parameters, depending on the network policy and it is dynamic.

The traffic prioritisation mechanism operates on the requirement for differentiating the traffic into different classes so that priorities can be assigned to those classes. The 802.1D standard specifies eight distinctive levels of priority (0-7) as in Table 1, each of which relates to a particular type of traffic. The transmitting station sets the priority of each packet. These priority bits are read and sorted enabling the packets to be forwarded to the appropriate buffers.

**Table 1: IEEE 802.1D Priority levels and traffic levels**

Priority Level	Traffic Type
0	Best effort
1	Background
2	Standard (spare)
3	Excellent effort (business critical)
4	Controlled Load (streaming multimedia)
5	Video, less than 100 milliseconds latency and jitter
6	Voice, less than 10 milliseconds latency and jitter
7	Network control reserved traffic

**Table 2 : Traffic Queue mapping to IEEE 802.1D**

Traffic Queue	IEEE 802.1D Priority Level	Traffic Type
0 (low)	0-2	Best Effort
1	3-5	Video and Business Critical
2	6	Voice
3 (high)	7	Network Control

Table 2 illustrates all traffic are mapped into 4 different priority buffers – very high priority (VH), high priority (H), low priority (L), and very low priority (VL). The classification of traffic type used in the mapping is based on the characteristic of applications. For example, the network control traffic is given the highest priority because of its importance to the network operations. Similarly with the traffic type of voice

which also requires higher priority, otherwise the quality of service will deteriorate (Wang 2001).

The drawback of this mechanism was that it does not address the issue of performance which is the real main concern in the networking. By using predefined rules in classifying the packets and without the consideration of their consequences to the possible delay accumulation, the final consequences could be devastating. Without the estimation calculation, the potential network performance could not be known. If one particular type of traffic is dominant and according to the policy it has to be channelled to the relevant buffer. This could result in the degradation of future performance of that particular buffer due to high occupancy. Delay has direct relationship with the buffer occupancy. For example, if the traffic of voice is constantly flowing in, the higher priority buffer ('H') would be full occupied. This would lead to the most recent packet that entering the buffer has to wait longer before has a chance to be popped out especially for a case that the scheduler uses FIFO algorithm. Obviously packets enter the buffers in sequential manner that is in FIFO fashion. However, the way of packets are despatched out is governed by the robin-robin algorithm.

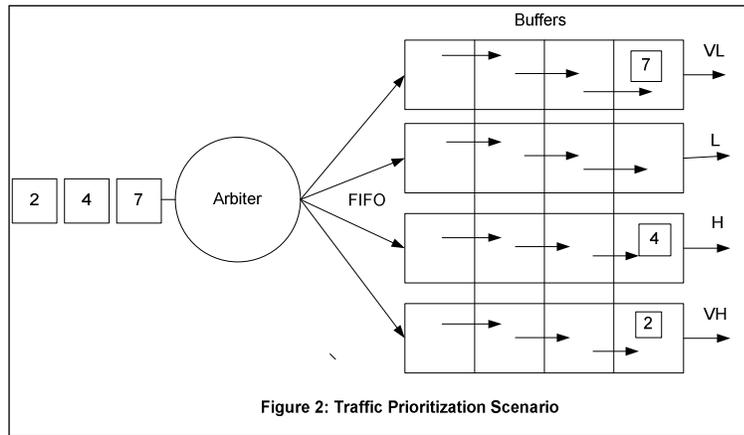


Figure 2: Traffic Prioritization Scenario

The arbiter that classifies the packets operates on the rule of if-then which has the goal of filtering the packet's TOS and forwarding to the appropriate buffer class (Gani, Abouzakhar, & Manson, 2003). Three bits of TOS field in the IP packet enable such classification can be done. However, the problem arises when the arbiter and the scheduler were let to operate independently without the synchronisation between pushing

and popping the packets. The scheduler state that signals the buffer to pop out the packets was not taken into the consideration when deciding which buffer class the packet has to be forwarded to. These two components operate independently with the regard to the performance optimisation.

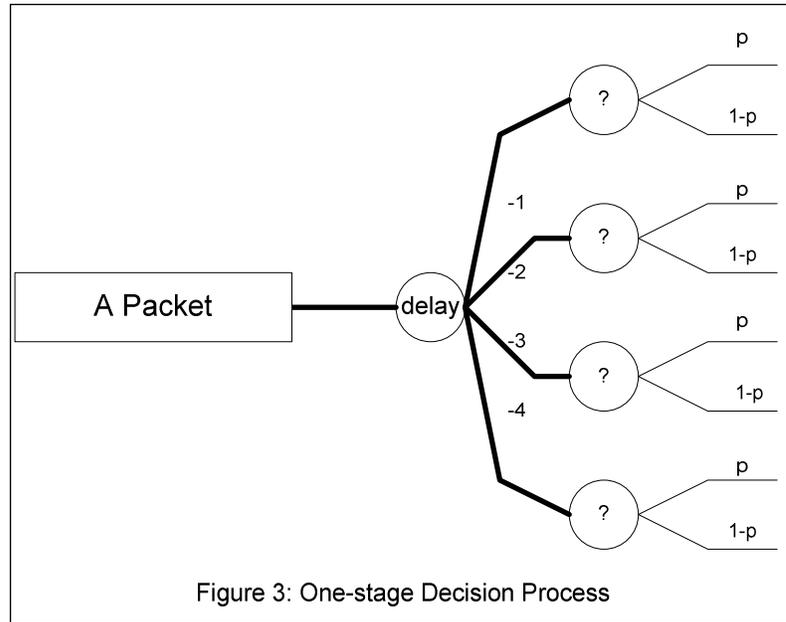
## The Methodology

We present our model with the goal of improving the buffer performance. Upon the arrival of packet, the arbiter (learner) has to make decision for classifying and forwarding the packets to the appropriate buffers. The decision is only considered a 'good' one if packets that are forwarded to

the buffer, are not delayed by the lower priority packets. Hence, information of the buffer states is essential in reaching such decisions. Figure 2 illustrates the scenario of packet arrival which requires the arbiter to decide so as to find the optimal policy  $\pi^*$  for that particular packet.

In formulating the traffic prioritisation process as an MDP, we present states of the model as the all possible combinations of buffer load and delay values. The transition of state to another is triggered by the action that was chosen randomly. For example, on the arrival of a packet, the learner has to compute the optimal delay time by performing an exploration that effectively constitutes a series of actions. However, the expected delay time of a packet is to be less than an average delay time with a probability  $p$  or stay the same with a probability of  $1-p$ .

Figure 3 illustrates one-stage decision process upon the arrival of a packet with higher priority tagging. Since it is a high priority packet, by default (specified in the predefined rules) it should be forwarded to the high priority buffer. The delay that it encounters would depend on the buffer load (the number of packets waiting in the buffer). If there are  $X_t$  packets inside the buffer at time  $t$ , a new arrival of packet will have to wait for all the packets in the buffer to be popped out. In other words, the delay has a direct



relationship with number of packets in the buffer. So the total delay  $\sum_{t=1}^n X_t$  that a packet has is the computation of every buffer class (VL, L, H, and VH) iteratively.

Depending on the transition probabilities ( $p$ ,  $1-p$ ) and the chosen action, the arbiter would be able to reach the best decision of a minimal delay time for a packet. Every action is associated with the reward which is represented by a negative enumeration. The arbiter chooses the action that can results in the maximum expected reward, based on the transition probabilities.

A Markov Decision Process evaluates the problem over the entire four buffer classes, identifies the action that minimize the expected cost. In exploring the possible future reward, the learner has to comply with the policy  $\pi$ . A decision rule that constitutes the policy was created at both points – the classifying and scheduling mechanism.

In order to validate the policy, we used mock-up data for the purpose of checking the transition probabilities that can produce an optimal reward. We created packet generator for that purpose using the Internet Protocol (IP) specifications.

## Result

The model was evaluated with an analysis and observation on the performance of delay policies. We first carried out an analysis to determine the delay that a packet will possibly experience by simulating our Java-coded switch which has four buffers. Delay was computed as the difference of time between packet at the packet generator and at the destination. Our computation of delay was based on the propagation and queue delays only.

Figure 4 shows the relationship between delay and buffer occupancy. It shows that the delay decreases when the buffers are heading to full occupancy. The graph also shows that the size of service contributes significantly towards the decrease of delay.

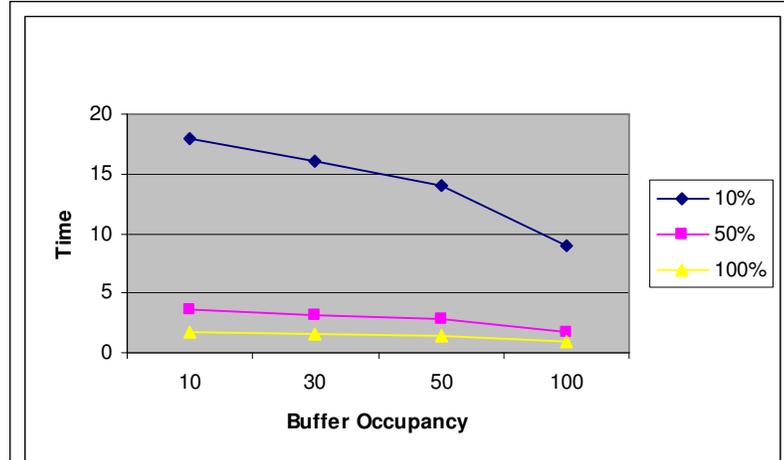


Figure 4: Delay Vs Buffer Occupancy

The optimal policies were applied to the test dataset and the distribution of delay was compared. The value function that represents the delay was computed iteratively so as to find the optimal rewards. The optimal policies which represent the optimal rewards provide some encouraging results. From the observation of delay distribution, it reveals that the delay time drops by 33% which means packets are delivered faster.

Figure 5 illustrates delay distribution for 100 trials. The maximum delay occurs at the beginning and gradually is getting better towards the end. This conforms to the Poisson distribution of traffic behaviour. The occurrence of delay with the small values towards the end shows that buffering had been improved.

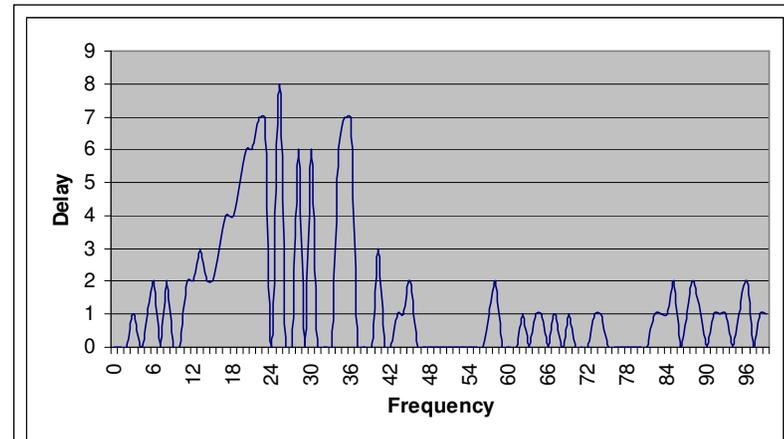


Figure 5: Buffering Performance

In addition, the model allows us to evaluate various performances such as transmission links, routing algorithms and routing and switching devices with the integration learning capabilities for improvement purposes.

## Conclusion

This paper showed that the network resource performance can be further improved by the deployment of MDP technique through the minimisation of delay that is associated with buffering. In this regard, optimisation is achieved through the minimisation of delay occurrences and indirectly improving throughput of the buffers. Thus, the movement of packet to its final destination

can be efficiently improved especially at the router/switch by optimising the delay time with the prediction of future delay occurrences.

The Markov Decision Process model also provides an insight on how complex systems such as network can be studied for a better understanding to achieve optimisation. This is because MDP is a systemic modelling tool that allows interactions between components are modelled statically as well as dynamically.

The Markov Decision Process model can be further enhanced with the integration of learning capability. Despite many techniques of learning are available such as Dynamic Programming and Monte Carlo, Q-Learning techniques was said to be the best one because it can work with the absence of perfect information. The Q-Learning addresses the issues of finding the transition probabilities for maximising the accumulative rewards through the prediction of future rewards.

It is learnt that MDP is a versatile idea that can be applied into many domains including network resource management and particularly for controlling and improving performance purposes. However, the deployment of MDP is hindered by the difficulties in presenting the problem as a Markovian system.

## References

- Bertsekas, D., & Gallager, R. (1991). *Data Networks* (2<sup>nd</sup> ed.). Prentice Hall.
- Gani, A., Abouzakhar, N., & Manson, G. (2003). Implementing traffic prioritisation in packet switches using intelligent arbiter. *PostGraduate Symposium (PGNet 2003)*, John Moore University, Liverpool, UK.
- Hardy, W. C. and NetLibrary Inc. (2001). *QoS measurement and evaluation of telecommunications quality of service*. Chichester, New York: Wiley.
- Howard, R.A. (1966). *Dynamic programming and Markov process*. MIT Press.
- Puterman, M.L. (1994). *Markov decision process: Discrete stochastic dynamic programming*. Wiley.
- Ratitch, B., & Precup, D. (2002). Characterizing Markov decision process. In *13<sup>th</sup> European Conf. on Machine Learning (ECML'02)*, Helsinki, Finland.
- Sharda, N. (1999). Multimedia Networks: Fundamentals and Future Directions. *Communications of Association for Information Systems, 1*, Article 10.
- Sutton, R.S. & Barto, A.G. (1998). *Reinforcement learning – An introduction*. Cambridge, MA: MIT Press.
- Tijms Henk, C. (1994). *Stochastic models: An algorithmic approach*. Chichester: Wiley.
- Wang, Z. (2001). *Internet QoS : Architectures and mechanisms for quality of service*. San Francisco & London: Morgan Kaufmann.
- White, D. J. (1978). *Finite dynamic programming: An approach to finite Markov decision processes*. Chichester: Wiley.

## Biographies

**Abdullah Gani** obtained B.Phil and M.Sc (Information Management) from Hull University in 1989 and 1990 respectively. He is currently a PhD candidate at the Dept of Computer Science, University of Sheffield, UK. His research domain is intelligent network resource management. He has published a number of papers related to network management areas. He has interest in education too especially related to the deployment of Information Technology in education.

**Omar Zakaria** obtained his B Comp Sc from Faculty of Computer Science & Information Technology, University of Malaya, Malaysia in 1994, and his MSc in Information Security from the

Information Security Group, Royal Holloway, University of London, UK in 1996. He is currently a PhD candidate at the Information Security Group, Royal Holloway, University of London, UK. His research area is information security management. He has published a number of papers related to information security areas.

**Nor Badrul Anuar Jumaat** obtained his Master of Computer Science from University of Malaya in 2003. Currently, he is a Lecturer at the Faculty of Computer Science and Information Technology in University of Malaya, Kuala Lumpur. His research interests include Computer Network Security, Open Source and IS/ICT.